

Refinement of the HEPAR Expert System: Tools and Techniques*

Peter Lucas
Department of Computer Science
Utrecht University
Padualaan 14
3584 CH Utrecht, The Netherlands

June 26, 2013

Abstract

Methods and tools for the static and dynamic verification and validation of software systems are common place in the field of software engineering. In the field of expert systems, where it is more difficult to ensure that a system meets the specifications and expectations than in traditional software engineering, such tools are generally not available. In this paper, the need for more support of the development process by method and tools is illustrated by the approach taken in building the HEPAR system, a rule-based expert system that can be used as a supportive tool in the diagnosis of disorders of the liver and biliary tract. At a certain stage in the development of this system an incremental development methodology has been adopted, in which implementation of parts of the expert system was followed by dynamic validation. For this purpose, a collection of software tools were implemented as extensions to a rule-based expert-system shell. These tools provide valuable information about the effects of modification of the HEPAR knowledge base, and indicate places in the knowledge base for refinement. It is believed that similar software tools may prove helpful in the development of other expert systems as well.

Keywords. Knowledge engineering; validation of expert systems; refinement of expert systems; evaluation of expert systems; expert-system validation tools.

1 Introduction

In the field of software engineering it is generally recognized that the implementation of large software systems must be supported by methods and tools for their verification and validation [19]. Often a distinction is made between static and dynamic approaches to verification and validation. Typical examples of *static* methods are program code inspection and methods for proving program correctness. The application of static methods does not require the program to be executed. In contrast, *dynamic* verification and validation methods involve execution of the program and examining its output when it is presented with certain input. As has been pointed out repeatedly by many researchers, dynamic methods can be used only to demonstrate the presence of errors in a program, and never to demonstrate their absence.

*Published in: *Artificial Intelligence in Medicine*, 6(2), 175–188, 1994.

Despite this fundamental limitation, software engineers consider dynamic methods indispensable aids in the software-development process, and supporting software tools are invariably included in programming environments. It is therefore ironical that in the development of expert systems, where it is much more difficult to ensure that the system meets its specifications and expectations than in software engineering, tools that aid in the dynamic verification and validation of the system are not generally available.

In this paper we discuss several static and dynamic methods, including some software tools, which were applied during the development of the HEPAR system, a rule-based expert system in the field of hepatology that offers support in the diagnosis of disorders of the liver and biliary tract. For a description of this system and of two successive studies of its performance, the reader is referred to [10, 11, 12]. Taking the development of the HEPAR system as a real-life example, we describe where in the development process the need for static and dynamic verification and validation methods and tools arose. This experience provides further evidence that in the development process of expert systems more support by methods and tools tailored to verification and validation is needed than is presently offered.

The structure of the paper is as follows. In Section 2, we review the design and implementation of the HEPAR system and analyse the problems encountered in the design stage. Several static verification and validation methods were employed at this stage of the project. In Section 3, the subject of knowledge-base refinement is related to expert-system validation using dynamic techniques. In Section 4, we describe some simple software tools which were used for the purpose of dynamic validation during the refinement of the HEPAR system.

2 Development of the HEPAR system

2.1 Knowledge acquisition and design

It is now well-recognized that the acquisition of domain knowledge in the process of building an expert system is a difficult task [6]. In recent years, many methodologies have therefore been proposed, providing systematic methods to be followed in building an expert system. Examples of such methodologies are KADS [3] and KEATS [15, 16]. Some of these methodologies include a set of software tools which help the knowledge engineer in building a specific application, mainly by assisting in the analysis of the problem domain. Some assistance by software tools may be provided in the design of the expert system as well. Examples of such tools are Shelley [2] and Acquist [15, 16]. Most methodologies place considerable emphasis on the process of gathering domain knowledge to be incorporated into the expert system, and on the development of conceptual models of the domain, being the result of the analysis of the knowledge collected.

Although the HEPAR system was actually designed long before such methodologies came into play, the development of the system was initially carried out in a structured way, mainly following the top-down design methodology from software engineering. The knowledge concerning diagnosis in liver and biliary disease incorporated into the HEPAR system was derived from the experience of a specialist in internal medicine and hepatology and from the medical literature. The analysis of the problem of diagnosis of disorders of the liver and biliary tract indicated that the following aspect were important in this domain:

- Expert hepatologists follow a clear and unambiguous strategy in diagnosis. The early classification of a patient's disorder into rather general categories, such as whether or not

the disorder is biliary-obstructive in nature, is used for the selection of supplementary tests to reduce the number of alternative diagnoses to be considered.

- Early in the diagnostic process only a limited amount of patient data is available, mainly obtained from medical history and physical examination. Still, a hepatologist is often capable of coming up with a working diagnosis of the patient's disorder.

In the design of the HEPAR system, the strategy followed by the hepatologist has been taken as the point of departure for problem decomposition of the diagnostic process, by distinguishing several subtasks [11].

The requirement that the ultimate system ought to be able to assist the clinician in the initial assessment of the patient, in whom only a limited amount of data is available, as well as in the assessment of the patient in whom more specific test results are known, proved to be extremely difficult. In general, to explicitly deal with data not available in the patient would yield an exponential number of combinations of conditions on known and unknown patient findings to be taken into account. Although the hepatologist involved in the project was able to reduce the number of useful combinations considerably, we felt quite uncertain with respect to the suitability of this knowledge for classifying actual patient cases.

Note that current knowledge-acquisition methodologies do not offer much help in solving this problem. In most popular methodologies, the design process is essentially viewed as the process of abstraction from reality. Our problem was that we required some form of experimental feedback in refining the expert system to accommodate to reality. Likewise, only limited attention has been given to tools that provide information about the diagnostic quality of the advice produced by the expert system.

2.2 Implementation and experimental feedback

Implementation of the HEPAR system was carried out using the EMYCIN-like expert-system shell DELFI-2 [9]. The advice produced by the system is in the form of a collection of conclusions:

1. Whether the patient has a hepatocellular or biliary-obstructive disorder.
2. Whether the features found in the patient indicate benign or malignant disease.
3. A differential diagnosis consisting of a collection of specific disorders, selected out of a set of about 80 disorders, each explaining some of the findings observed in the patient.

In HEPAR, the first two conclusions are intermediate and the last is a final conclusion.

After completing a considerable portion of the knowledge base, it was decided to carry out some experiments with the system using data from real patients to investigate whether the system was able to meet our expectations. It appeared that the system was unable to come up with an acceptable advice in many cases. An analysis of the results of this initial experiment yielded the following reasons for the disappointing performance:

- Many rules were formulated too rigorously, such that these rules almost never applied in a patient with the given disease.
- Many rules were defined without explicitly mentioning the medical context in which they should hold. These rules frequently succeeded in patients for whom they had not been designed.

```

if
  same(patient,complaint,abdominal_pain) and
  ⇒ notsame(patient,complaint,fever) and
  same(patient,signs,hepatomegaly) and
  same(pain,character,continuous) and
  same(ultrasound_liver,parenchyma,multiple_cysts) and
  same(patient,nature-disorder,benign)
then
  conclude(patient,diagnosis,polycystic_disease)
  with CF = 1.00
fi

```

Figure 1: Weakly formulated production rule.

```

if
  (same(complab,duration,chronic) or
  same(patient,type-disorder,biliary_obstructive)) and
  same(patient,sex,female) ⇒ same(patient,sex,male) and
  same(serol,mitochondrial_Ab,yes)
then
  conclude(patient,diagnosis,primary_biliary_cirrhosis)
  with CF = 0.80 ⇒ with CF = 0.60
fi

```

Figure 2: Rigorously formulated production rule.

These problems may actually be taken as an indication of the knowledge clinicians draw upon in medical practice. Firstly, the knowledge of the clinician is partly based on the descriptions given in medical textbooks, in which there is little place for the description of atypical disease patterns, and partly on experience in the management of specific disorders. Rules which have been formulated too rigorously tend to describe the typical picture of the disease, and may assume the availability of an unrealistic amount of data for the patient. Secondly, the clinician has considerable experience with disorders frequently observed in clinical practice. However, these disorders carry a clinical context which the clinician may not be able to make explicit. Formalizing such knowledge may yield rules with a wider application than intended.

As an example, consider the production rule depicted in Figure 1. In this HEPAR rule we use the object–attribute–value representation and certainty factors as employed in the DELFI-2 system [9, 13]. This rule was originally formulated without the second condition (indicated by the right arrow); for the modified rule to be applicable, fever must be absent in the patient. So, in its original form it is an example of a too weakly formulated production rule. This missing condition caused the original rule to interact with Caroli’s disease, infected liver cysts and cystic liver metastases.

The production rule shown in Figure 2 is an example of a too rigorously formulated rule, since in the form left from the right arrows, the rule is only applicable for female patients. Typically, a patient having primary biliary cirrhosis is a female, but the disorder is not limited to the female sex. A new production rule was therefore added to the knowledge base in which the expressions specified right from the arrow replaced the expressions left from the arrow.

As said above, clinicians often have to base their early decisions on incomplete clinical evidence; likewise, expert systems must be able to deal with incomplete evidence as well. In designing and implementing the HEPAR system we have tried to explicitly handle incomplete patient data in the following two ways:

1. By distinguishing several different conceptual levels in the diagnostic problem-solving process;
2. By explicitly incorporating knowledge about unknown diagnostic test results for a patient into the knowledge base.

To deal with the first source of incompleteness of information in the HEPAR system, rules were drafted covering only the symptoms and signs of a disorder obtained early in the diagnostic process, whereas other rules were drafted only covering the results of supplementary tests obtained later in the diagnostic process. In this way a more or less layered structure of the knowledge base was obtained. The second source of incompleteness was dealt with by inspecting rules for conditions on data not always available in the patient. Some of these rules were used as a basis for new rules containing conditions concerning unknown data. Mainly static verification and validation methods were employed at this stage; the early experiments were only carried out to validate our design rationale.

3 Knowledge base refinement and validation

3.1 Refinement parameters

The process of refining an expert system may be viewed as the iterative process of validating, extending and adapting its knowledge base. Here, we are concerned with dynamic validation. Since the extension and adaptation are based on the results of the validation, it is clearly important to decide on the parameters used for validating an expert system. The point of departure for expert system validation is to consider a medical diagnostic expert system as a computer program that tries to construct a model of a given patient which it compares with prestored descriptions of disease patterns in its knowledge base. Under ideal circumstances, validation of a diagnostic expert system should therefore not only pertain to the advice produced by the expert system, but also to the assumptions made by the reasoning process on which the conclusions are based. It is, for instance, important to know whether it is possible for an expert system to arrive at conclusions which are correct, but based on incorrect assumptions. So, the conclusions of a medical diagnostic expert system should not be interpreted as unique answers, but as judgements of the patient's status [8]. As a consequence, not unique answers but judgements should be validated. This view of validation is also able to cope with situations in which the expert system's conclusions are incorrect, but nevertheless acceptable in the light of the data available. This approach to validation is particularly valuable when applied in refining an expert system; it is less appropriate in the final validation of an expert system [7].

3.2 Refinement by dynamic validation

For the purpose of the refinement of the HEPAR knowledge base, we have investigated several approaches to system validation. A diagnostic expert system like HEPAR may be validated against:

1. Patient cases with known clinical diagnosis;
2. The conclusions of some other, but similar decision support system;
3. The judgement of human experts in the field.

In all three cases, there is a need for a test, or a combination of tests, that may be taken as a 'gold standard' for the comparison, although this is not as crucial as in the final validation of an expert system, because inspection of the knowledge base may provide additional information. Examples of tests that are suitable as a gold standard in the diagnosis of disorders of the liver and biliary tract are the histological examination of liver biopsies, ERCP and surgical exploration. There is not a single test available in hepatology that may be employed as a gold standard in the entire domain, because diagnosis of hepatocellular disorders differs considerably from diagnosis of biliary obstruction.

Initially in the refinement, we have used part of the data from more than 1000 patients obtained from the Danish COMIK group as a source for comparison. Originally, these data have been used in the development of the Copenhagen Pocket Chart, a paper chart based on the statistical technique of logistic regression, which may assist the clinician in the early assessment of a patient with jaundice [14]. Unfortunately, this database was of limited value because only 23 disease categories were distinguished in this database, whereas in HEPAR more than 80 disorders of the liver and biliary tract are distinguished, and not all data required for HEPAR to derive final conclusions were included in the database. Therefore, the database was mainly useful for getting insight into the extent to which HEPAR was capable of dealing with incomplete patient data. During the further development of the system, a database with patient data from the Leiden University Hospital was put together, which was applied as the main device for the refinement of the knowledge base.

Comparison of an expert system with some similar decision-support system is seldom straightforward, because of differences in required input and produced output among the systems. For the refinement of the HEPAR system, we have included production rules that map diagnostic conclusions of HEPAR to the possible diagnostic conclusions of the Copenhagen Pocket Chart. The results produced by the Copenhagen Pocker Chart could thus be used as a simple means for rapidly finding patient case which deserved further study with regard to the conclusions produced by HEPAR.

The hepatologist involved in the project has studied the reasoning process of HEPAR for a considerable number of patients. Validation of the reasoning process of HEPAR turned out to be very time-consuming, and as a consequence we have not been able to involve other hepatologists. However, on a limited scale we have profited from discussion with other hepatologists in examing the results produced by HEPAR.

Taking the conclusions concerning the final diagnosis produced by the HEPAR system as a point of departure for refining the system was difficult, because the system's advice consists of more than one conclusion. This problem is known as the multiple response problem [18]. Consider for example the situation that the expert system did produce the correct answer with highest ranking, as well as a conclusion with lower ranking which is however totally

unacceptable to the clinician. Restricting the validation only to the single conclusion with the highest ranking will give a distorted account of the actual performance of the system. These problems cannot be solved by only collecting information concerning the number of conclusions generated by the system and the ranking of the clinical diagnosis. Furthermore, consider the situation in which the conclusion with highest ranking is incorrect, but where the differential diagnosis as a whole is acceptable to the clinician. Taking only the conclusion with highest ranking into account will then give an inadequate impression of the system's capabilities. However, due to the layered approach to diagnostic problem solving modelled in HEPAR, it is not only possible to compare specific disorders as a diagnosis with the clinical diagnosis confirmed in the patient, but also to check whether the patient's disorder has been classified into the right diagnostic category (e.g. hepatocellular disorder). This layered approach makes it less likely that the differential diagnosis produced by the system is as a whole unacceptable to the clinician.

Most of the information obtained by the dynamic validation of the HEPAR knowledge base was automatically compiled by a collection of simple software tools which are discussed in the next section. Without the availability of these tools, dynamic validation would have been too time-consuming for being practically feasible.

4 Tools for knowledge base refinement

4.1 Testing tools in software engineering

Some of the software tools that have been developed for the dynamic validation of the HEPAR system, and applied in the refinement and the performance studies of the system, have been inspired by software tools commonly available in programming environments.

Test-data generators, programs that systematically produce test data to be used as input to the program to be tested, are one type of program that may be useful in the dynamic validation of expert systems. However, for realistic testing, data from real-life cases are often indispensable.

Another tool is the *dynamic analyser*, also known as the execution flow summarizer. A dynamic analyser adds instrumentation statements to a computer program in order to collect information on how many times a statement is executed. A display part of the dynamic analyser prints a summarizing execution report [19]. A tool with similar usage as the dynamic analyser is the *call-graph profiler* [5].

4.2 A tool for performance measurement

The environment of software tools developed for the dynamic validation of HEPAR consists of a non-interactive batch version of the expert-system shell DELFI-2 which is able to use a database of patient cases as its input. This system produces a report containing the results for each individual patient. The report together with a file containing information about the final clinical diagnoses and the two intermediate conclusions concerning the patient is then processed by a program which produces a table summarizing the results. Figure 3 shows the overall structure of the validation environment. The tools collect information with regard to the number of correct, incorrect and unclassified patient cases concerning:

1. the type of hepatobiliary disease (hepatocellular or biliary-obstructive);

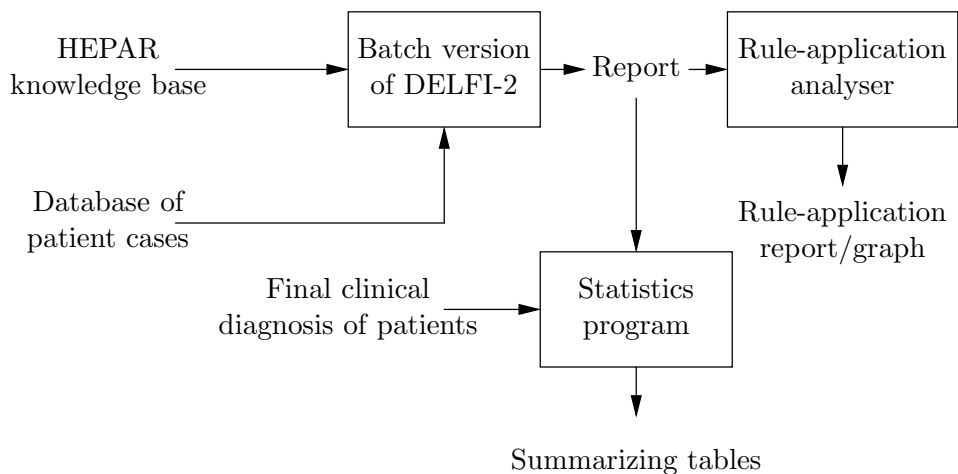


Figure 3: Environment for dynamic validation.

2. the nature of the disorder (benign or malignant);
3. the final diagnosis.

With regard to the final diagnosis, the system computes the average number of conclusions, whether the clinical diagnosis occurs as the conclusion ranked first, or among the list of alternatives generated. An example of such a table, produced after refinement of the HEPAR system using the software tools and a database with data from 82 patients from Leiden University Hospital, is reproduced in Table 1, which shows the results for the patients after refinement. The reader should note that the system does not reach 100% correctness, for reasons discussed in Section 3.1.

To obtain some insight into the capabilities of the system in handling incomplete data, the batch version of the expert-system shell can select part of the patient data from the database. For example, the system is capable of selecting only data obtained from history and physical examination. An example of a table in which the results for incomplete data are produced using this environment is shown in Table 2. As can be seen in Table 2, the number of correct final diagnoses decreased considerably when the patient data entered into the system were more incomplete. However, the percentage of incorrectly classified cases did not increase

Table 1: Diagnostic results of HEPAR for a population of 82 patients with hepatobiliary disease.

Conclusion	Correct	Incorrect	Unclassified	Total
	<i>n</i> (%)	<i>n</i> (%)	<i>n</i> (%)	<i>n</i> (%)
Type of hepatobiliary disorder	74 (90)	4 (5)	4 (5)	82 (100)
Benign/malignant nature of disorder	78 (95)	4 (5)	0 (0)	82 (100)
Final diagnosis	71 (86)	8 (10)	3 (4)	82 (100)
Clinical diagnosis among conclusions	76 (93)	3 (4)	3 (4)	82 (100)

Table 2: Assessment of the effects of incompleteness of information on the diagnostic conclusions of the system, for a database of 82 patients with hepatobiliary disease.

Conclusion	Correct (%)			Incorrect (%)			Unclassified (%)		
	A	B	C	A	B	C	A	B	C
Type of hepato-biliary derangement	90	90	50	5	5	0	5	5	50
Benign or malignant nature of disorder	95	95	94	5	5	6	0	0	0
Final diagnosis	86	49	24	10	9	10	4	43	66

A: All available data presented to system.

B: Only data concerning symptoms, signs, haematology and bloodchemistry (no data from ultrasound or serology presented).

C: Only data from medical interview and physical examination.

significantly, only the percentage of unclassified cases did. Tables such as presented here were used by the hepatologist as an indication of the effects of changes to the knowledge base. An accompanying textual report provided information for each individual patient, and gave also information about the rules applied in deriving the final conclusions. This information served as a point of departure for a more in-depth study of the reasoning behaviour of the system.

4.3 Dynamic analysis of the HEPAR knowledge base

In the previous section, we have discussed how the study of the results of the HEPAR system for individual patient cases, has been employed for refining the diagnostic quality of HEPAR. A second source of information that has been used for the refinement of the system, was the contents of the HEPAR knowledge base itself, by studying the overall behaviour of the system when provided with a complete database with patient data. These tools bear some resemblance with a dynamic analyser as described in Section 4.1. In order to obtain information concerning the frequency of rule application over a given database of patient cases, the testing environment discussed in Section 4.2 includes a collection of tools which uses the report produced by the batch version of the expert system for a database of patient cases. The following results are produced by these programs:

- An enumeration of all production rules used, with for each rule information about how often it has been used for a given database;
- An overview of the frequency distribution of the rule application, both in textual and in graphical form.

Figure 4 which was automatically produced by the environment, contains the results after refinement of the HEPAR knowledge base for the 82 patient cases from Leiden University Hospital. Most production rules (72 of about 500 rules contained in HEPAR) were applied only once. The accompanying textual form, which is reproduced in Figure 5, shows that from the rules that were applied several times, those with highest frequency were applied to conclude about the intermediate hypotheses. Only few, one to three, production rules were applied several times to reach a final conclusion. Because we have tried to obtain a rulebase

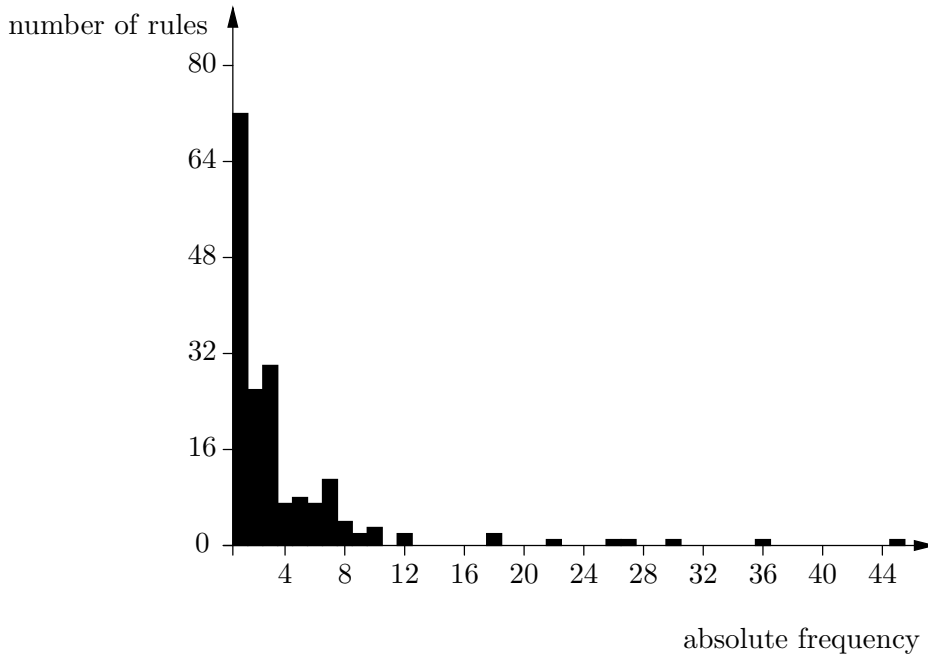


Figure 4: Rule-application bar graph for 82 patients.

in which at most a few rules will succeed for a given case, the report does not provide results concerning failed rules. The reports were studied by the hepatologist involved in the project as another source for the refinement of HEPAR.

5 Discussion

Recent knowledge-engineering methodologies place considerable emphasis on the development of conceptual models. A suitable conceptual model may be of real help in designing and implementing an expert system, as was also observed in the development of the HEPAR system, where a diagnostic problem-solving strategy was used as the basis for problem decomposition and structuring of the knowledge base. However, in many fields of medicine, the development of an expert system is only possible with sufficient experimental feedback, for which software tools are required. Some other software tools supporting the building of rule-based expert systems have been developed in the past. Teiresias was an experimental tool that assisted in the refinement of rule-based expert systems by interacting with the user in the analysis of the conclusions concerning single cases, applying meta-knowledge about the rulebase [4]. Although such an analysis is certainly useful, an approach as embodied by Teiresias does not give information about how well the system performs over a database of cases. Seek is a system that automatically suggests generalizations and specialization of production rules, based on the analysis of the success and failure of rules on processing case data [17]. This system is more in the line of our approach. The broadness of the domain of hepatology, and the amount of patient data incorporated in the HEPAR system, suggest that automatic techniques as provided by Seek are as yet not powerful enough to be applied for refining a system like HEPAR. The parameters used as a point of departure for the refinement of the HEPAR knowledge base are only a few of the many that are possible. Another elegant example of

FREQUENCY	#RULES	RULE NUMBER
1	72	161, 500, 510, 660, 700, 720, 790, 800, 860, 900, 930, 950, 980, 1100, 1140, 1150, 1330, 1340, 1370, 1410, 1480, 1540, 1610, 1630, 1710, 1730, 1820, 1980, 2000, 2010, 2030, 2050, 2140, 2240, 2250, 2380, 2530, 2680, 2690, 2750, 3011, 3020, 3022, 3090, 3100, 3101, 3120, 3160, 3171, 3192, 3220, 3340, 3370, 3425, 3430, 3450, 3460, 3470, 3520, 3530, 3610, 3780, 3820, 3870, 3902, 3930, 3960, 4000, 4111, 4122, 4162, 4260
2	26	110, 120, 130, 440, 460, 470, 740, 830, 1090, 1130, 1490, 1940, 1950, 2130, 2180, 2370, 3010, 3041, 3110, 3390, 3410, 3427, 3428, 3440, 3920, 3940
3	30	111, 112, 131, 132, 140, 190, 230, 240, 560, 580, 600, 620, 1220, 1900, 1970, 2060, 2080, 2110, 2170, 2220, 2280, 2310, 2720, 3190, 3350, 3970, 4230, 4240, 4250, 4252
4	7	260, 341, 890, 1580, 2160, 3080, 4320
5	8	1550, 1560, 2120, 2350, 3001, 3060, 3102, 3420
6	7	160, 300, 2330, 3050, 3070, 3900, 3903
7	11	90, 100, 332, 1210, 1570, 2200, 2320, 2340, 2970, 3000, 3910
8	4	270, 290, 340, 350
9	2	150, 180
10	3	250, 4080, 4310
12	2	540, 3103
18	2	220, 4020
22	1	370
26	1	5010
27	1	5000
30	1	5030
36	1	711
45	1	5020

Figure 5: Textual rule application form.

knowledge-base refinement has been proposed by Adlassnig and Scheithauer in the context of the CADIAG-2/PANCREAS system [1]. They have used ROC curves for the optimal adjustment of the internal classification threshold in this expert system. The technique is, however, not applicable in the HEPAR system, because here failure of classification is the result of logical falsification, and not of the failure of reaching an internal threshold.

Most current expert-system shells and expert-system builder tools do not offer facilities supporting the refinement of an expert system by dynamic validation. In the development of the HEPAR system, we have therefore developed a collection of simple software tools that provide useful information for the refinement of the system. These tools have also been used in two successive final validation studies of HEPAR [10, 12]. Although there are many ways in which these tools can be improved, these validation studies would not have been possible without the assistance of these tools. In our opinion, future software tools for building expert systems should offer a wider range of facilities for the detailed analysis, verification and validation of an expert system than is currently provided.

References

- [1] K.P. Adlassnig and W. Scheithauer, Performance evaluation of medical expert systems using ROC curves, *Computers and Biomedical Research* 22 (1989) 297-313.
- [2] A. Anjewierden, J. Wielemaker, C. Toussaint, Shelley – Computer aided knowledge engineering, in: B. Wielinga, J. Boose, B. Gaines, G. Schreiber and M. van Someren, eds., *Current Trends in Knowledge Acquisition* (IOS Press, Amsterdam, 1990) 41-59.

- [3] J. Breuker and B. Wielinga, Models of expertise in knowledge acquisition, in: G. Guida and C. Tasso, eds., *Topics in Expert Systems Design* (North-Holland, Amsterdam, 1989) 265-295.
- [4] R. Davis and D.B. Lenat, *Knowledge-based Systems in Artificial Intelligence* (McGraw-Hill, New York, 1982).
- [5] S.L. Graham, P.B. Kessler and M.K. McKusick, Gprof: a call graph execution profiler, in: *Proceedings of the SIGPLAN'82 Symposium on Compiler Construction*, SIGPLAN Notices 17 (1982) 120-126.
- [6] G. Guida and C. Tasso, Building expert systems: from life cycle to development methodology, in: G. Guida and C. Tasso, eds., *Topics in Expert Systems Design: methodologies and tools* (North-Holland, Amsterdam, 1989) 3-24.
- [7] J. Hilden and J.D.F. Habbema, Evaluation of clinical decision aids – more to think about, *Medical Informatics* 15 (1990) 275-284.
- [8] C.A. Kulikowski and S.M. Weis, Representation of expert knowledge for consultation: the CASNET and EXPERT projects, in: P. Szolovits, ed., *Artificial Intelligence in Medicine* (Westview Press, Boulder, 1982) 21-56.
- [9] P.J.F. Lucas, *Knowledge Representation and Inference in Rule-based Systems*. Centre for Mathematics and Computer Science, Report CS-R8613, Amsterdam, 1986.
- [10] P.J.F. Lucas, R.W. Segaar, A.R. Janssens, HEPAR: an expert system for the diagnosis of disorders of the liver and biliary tract, *Liver* 9 (1989) 266-275.
- [11] P.J.F. Lucas, A.R. Janssens, Development and validation of HEPAR, an expert system for the diagnosis of disorders of the liver and biliary tract, *Journal of Medical Informatics* 16 (1991) 259-270.
- [12] P.J.F. Lucas, A.R. Janssens, Second evaluation of HEPAR, an expert system for the diagnosis of disorders of the liver and biliary tract, *Liver* 11 (1991) 340-346.
- [13] P.J.F. Lucas, L.C. van der Gaag, *Principles of Expert Systems* (Addison-Wesley, Wokingham, 1991).
- [14] P. Matzen, A. Malchow-Møller, J. Hilden, C. Thomsen, L.B. Svendsen, J. Gammelgaard, E. Juhl, Differential diagnosis of jaundice: a pocket diagnostic chart, *Liver* 4 (1984) 360-71.
- [15] E. Motta, T. Rajan and M. Eisenstadt, A methodology and tool for knowledge acquisition in KEATS-2, in: G. Guida and C. Tasso, eds., *Topics in Expert Systems Design* (North-Holland, Amsterdam, 1989) 297-322.
- [16] E. Motta, T. Rajan, J. Domingue and M. Eisenstadt, Methodological foundation of KEATS, the knowledge engineer's assistant, in: B. Wielinga, J. Boose, B. Gaines, G. Schreiber and M. van Someren, eds., *Current Trends in Knowledge Acquisition* (IOS Press, Amsterdam, 1990) 257-275.
- [17] P.G. Politakis, *Empirical Analysis for Expert Systems* (Pitman, London, 1985).

- [18] R.E. Shannon, *Systems Simulation: the art and science* (Prentice-Hall, Englewood Cliffs, New Jersey, 1975).
- [19] I. Sommerville, *Software Engineering* (Addison-Wesley, Wokingham, 1992).